



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Học máy cơ bản

Tiền xử lý dữ liệu

Đoàn Phong Tùng

Nội dung môn học

- Buổi 1: Giới thiệu về Học máy
- **Buổi 2: Tiền xử lý dữ liệu**
- Buổi 3: Hồi quy tuyến tính
- Buổi 4: Học dựa trên láng giềng gần nhất (KNN)
- Buổi 5: Cây quyết định và Rừng ngẫu nhiên
- Buổi 6: Naïve Bayes
- Buổi 7: Máy vector hỗ trợ (SVM)
- Buổi 8: Đánh giá hiệu quả của mô hình học máy
- Buổi 9: Phân cụm
- Buổi 10: Kiểm tra giữa kỳ và trình bày ý tưởng làm dự án cuối kỳ

Nội dung

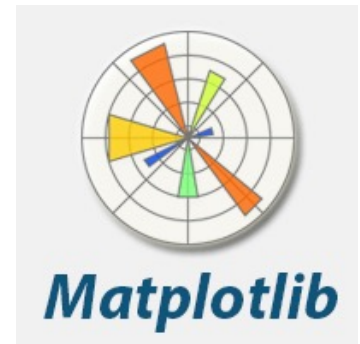
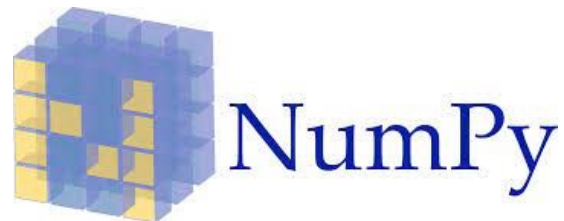
1. Giới thiệu về Scikit-learn, Keras/Pytorch
2. Quy trình xây dựng hệ thống học máy, khai phá dữ liệu
3. Thu thập và tiền xử lý

1. Giới thiệu về Scikit-learn, Keras/Pytorch



Giới thiệu thư viện Scikit-Learn

- **Scikit-learn**: Là thư viện mạnh mẽ về các thuật toán học máy và khai phá dữ liệu bằng ngôn ngữ Python
- Bắt nguồn từ một dự án của Google, Sklearn được xây dựng chủ yếu dựa trên Python, trên nền tảng của một số thư viện khác: NumPy, SciPy, Matplotlib, IPython, SymPy,



Giới thiệu thư viện Scikit-Learn

- Nhóm các thuật toán (https://scikit-learn.org/stable/user_guide.html):
 - Supervised learning
 - Unsupervised learning
 - Model selection and evaluation
 - Dataset transformations
 - Visualization
 - Dataset loading utilities
 - ...

Cài đặt Sklearn

- Trước khi cài Sklearn, cần cài trước các thư viện sau:
- Python (≥ 2.6 or ≥ 3.3),
- NumPy ($\geq 1.6.1$),
- SciPy (≥ 0.9).
- Sau đó, có thể dùng pip / conda gọi câu lệnh cài đặt sklearn:
 - *!pip install -U scikit-learn*
 - *!conda install scikit-learn*
- !!!Khi sử dụng, ta luôn cần lệnh ***import sklearn...***

Một số ví dụ

- Tiền xử lý: [sklearn.preprocessing](#)

```
from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
sc.fit(X_train)
```

- Chia dữ liệu train/test: [sklearn.model_selection](#)

```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```


Một số ví dụ

- Huấn luyện mô hình tuyến tính: `sklearn.linear_model`

```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
lr = LinearRegression()
lr_lasso = Lasso()
lr_ridge = Ridge()
```

- Huấn luyện và đánh giá:

```
lr.fit(X_train, y_train)
lr_score = lr.score(X_test, y_test) # with all num var 0.7842744111909903
lr_rmse = rmse(y_test, lr.predict(X_test))
lr_score, lr_rmse
```

(0.7837532911322952, 65.91685277030534)

Một số ví dụ

- Mô hình SVM và Rừng ngẫu nhiên

1.2.2 Support Vector Machine

```
from sklearn.svm import SVR
svr = SVR()
svr.fit(X_train,y_train)
svr_score=svr.score(X_test,y_test)
svr_rmse = rmse(y_test, svr.predict(X_test))
svr_score, svr_rmse
```

(0.24613512350275257, 123.07460910013376)

1.2.3 Random Forest

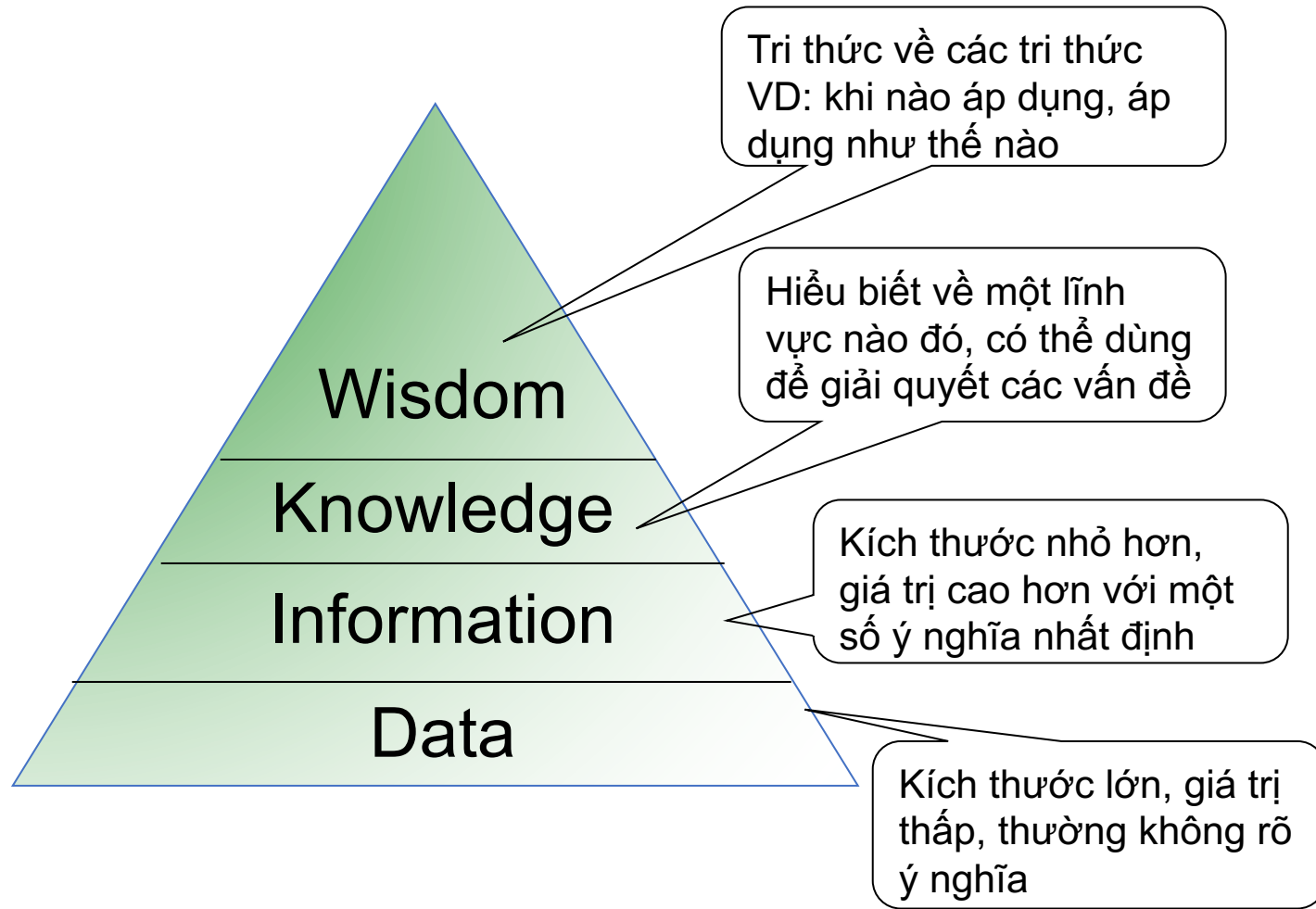
```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor()
rfr.fit(X_train,y_train)
rfr_score=rfr.score(X_test,y_test)
rfr_rmse = rmse(y_test, rfr.predict(X_test))
rfr_score, rfr_rmse
```

(0.8922988135841156, 46.51916964240559)

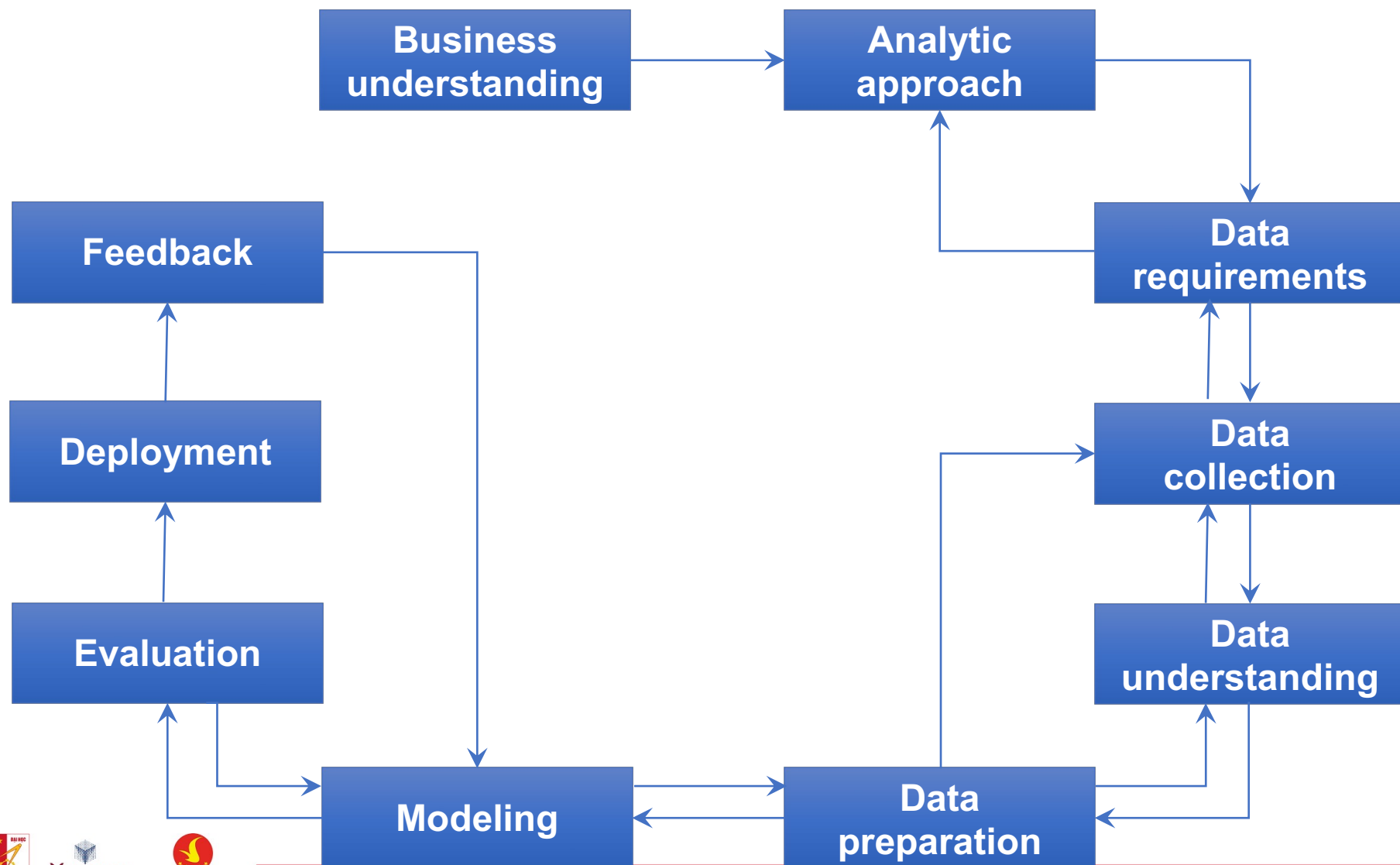
Keras/Pytorch

- Thư viện hỗ trợ lập trình giải bài toán học sâu: Keras, Pytorch, Caffee, TensorFlow, etc.
- ⇒ Giúp việc lập trình cho bài toán học sâu trở nên đơn giản và hiệu quả hơn.
- Học trong phần học sâu

2. Quy trình xây dựng hệ thống học máy, khai phá dữ liệu



Quy trình thực hiện: **hướng sản phẩm**



3. Tiền xử lý dữ liệu



Tại sao?

- Tiền xử lý để
 - Thuận tiện trong lưu trữ, truy vấn
 - Các mô hình học máy thường làm việc với dữ liệu có cấu trúc: vector, ma trận, chuỗi, ...
 - Hiệu quả của từng phương pháp học máy phụ thuộc rất nhiều vào cách **biểu diễn dữ liệu (data representation)**

Input

Mẫu dữ liệu thô (text, ảnh, audio, ...)

Output

Dữ liệu số theo từng ML/AI model(s)



$$x^{(n)} = \begin{bmatrix} -0.0920 \\ 3.4931 \\ -1.8493 \\ \dots \\ -0.2010 \\ -1.3079 \end{bmatrix}$$

$$\mathcal{D} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \dots \\ x^{(n)} \end{bmatrix}$$

Tính nguyên thủy của dữ liệu

Completeness (đầy đủ)

Từng mẫu thu thập nên đầy đủ thông tin các trường thuộc tính cần thiết

Integrity (trung thực)

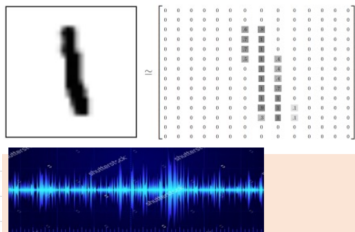
- Nguồn thu thập chính thống, đảm bảo mẫu thu được chứa giá trị chính xác trên thực tế.
- Jan. 1 as *everyone's* birthday? – *intentional (systematic) noises*

Homogeneity (đồng nhất)

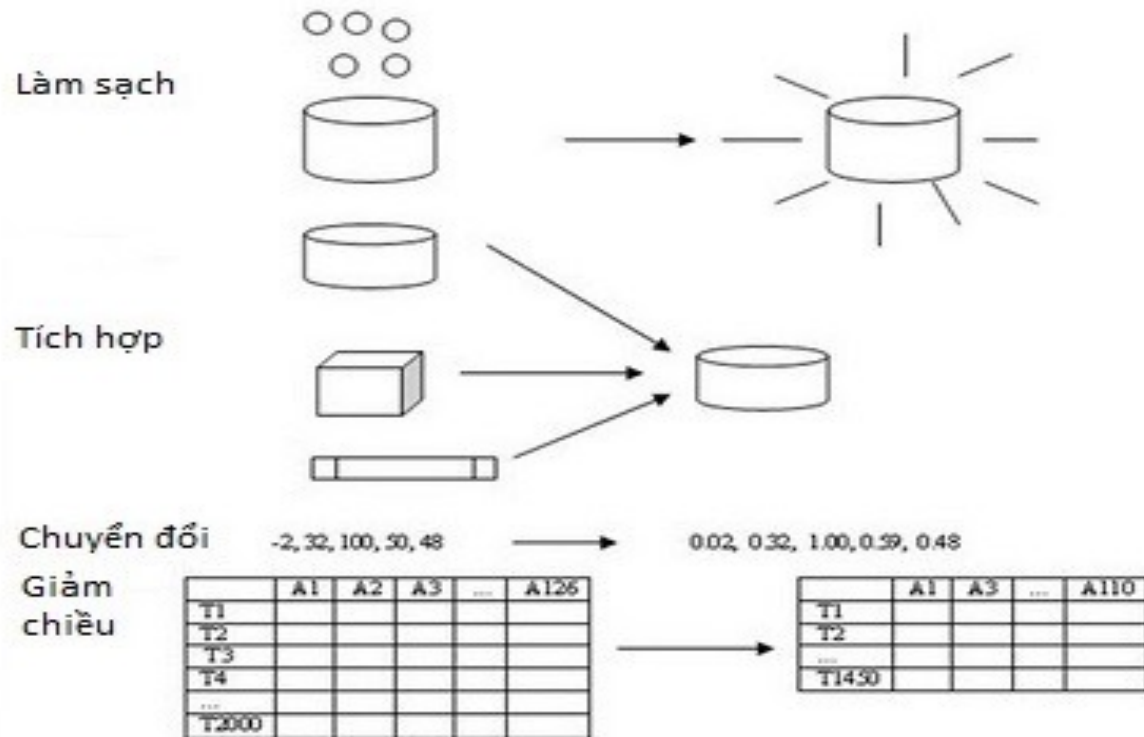
- Rating “1, 2, 3” & “A, B, C”; or Age = “42” & Birthday = “03/07/2010” (*inconsistency*)
- Heterogenous data sources / schemas

Structures (cấu trúc)

C	D	E	F
Population	Under15	Over60	Fertil
13724	40.24	5.68	3.64
14075	46.73	3.95	5.77
23852	40.72	4.54	4.35
90796	22.87	9.32	1.79
29955	28.84	9.17	2.44
247	37.37	6.02	3.46
28541	28.9	6.38	2.38
3395	22.05	18.59	2.07



Quy trình tiền xử lý dữ liệu



1. LÀM SẠCH DỮ LIỆU (Data cleaning)

- Đây là thủ tục quan trọng gồm ba bước chính
 1. Điền đầy các giá trị bị mất
 2. Loại nhiễu
 3. Kiểm tra và sửa tính không nhất quán



Làm sạch dữ liệu

- incomplete: Thiếu thuộc tính
 - e.g., *Occupation*=" " (missing data)
- noisy: Chứa noise, errors, or outliers
 - e.g., *Salary*="−10" (an error)
- inconsistent: Chứa dữ liệu khác nhau của cùng codes hoặc names, e.g.,
 - *Age*="42", *Birthday*="03/07/2010"
 - Was rating "1, 2, 3", now rating "A, B, C"
 - discrepancy between duplicate records

LÀM SẠCH DỮ LIỆU

- **Bước 1:** Điền đầy các giá trị bị mất, có thể chọn một trong các phương pháp
 - Bỏ không xét đến bộ dữ liệu bị mất giá trị
 - Điền lại giá trị bằng tay
 - Gán cho giá trị nhãn đặc biệt hay ngoài khoảng biểu diễn
 - Gán giá trị trung bình cho nó.
 - Gán giá trị trung bình của các mẫu khác thuộc cùng lớp đó.
 - Tìm giá trị có xác suất lớn nhất điền vào chỗ bị mất (**hồi quy, suy diễn Bayes, cây quyết định qui nạp**)

A1	A2	A3	A4	A5	A6	A7	A8	y
?	3.683	?	-0.634	1	0.409	7	30	5
?	?	60	1.573	0	0.639	7	30	5
?	3.096	67	0.249	0	0.089	?	80	3
2.887	3.870	68	-1.347	?	1.276	?	60	5
2.731	3.945	79	1.967	1	2.487	?	100	4

LÀM SẠCH DỮ LIỆU

- **Bước 2:** Loại nhiễu, có thể chọn một trong các phương pháp
 - Chia khoảng (Binning): Chia khoảng
 - Hồi quy (Regression) : sẽ dành chương riêng
 - Phân cụm (Cluster) : sẽ dành chương riêng

Loại nhiều: Chia khoảng

- Chia khoảng theo Equal-width (distance):
 - Chia miền dữ liệu thành N khoảng kích thước bằng nhau: uniform grid
 - Ví dụ: A và B là giá trị nhỏ nhất và lớn nhất của miền, độ rộng của 1 khoảng: $W = (B-A)/N$.
 - Lấy giá trị đại diện khoảng theo trung bình (mean)/ trung vị (median)/ biên (boundaries)
 - Nhược điểm: Không làm việc tốt với dữ liệu ngoại lai
- Chia khoảng theo Equal-depth (frequency):
 - Chia miền dữ liệu thành N khoảng mà tần suất xuất hiện của dữ liệu đều nhau.
 - Khó làm việc với thuộc tính dạng mục (categorical feature)

Loại nhiễu: Chia khoảng

Ví dụ:

Sắp xếp dữ liệu (đơn vị dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Chia vào các khoảng (equi-depth) bins:

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

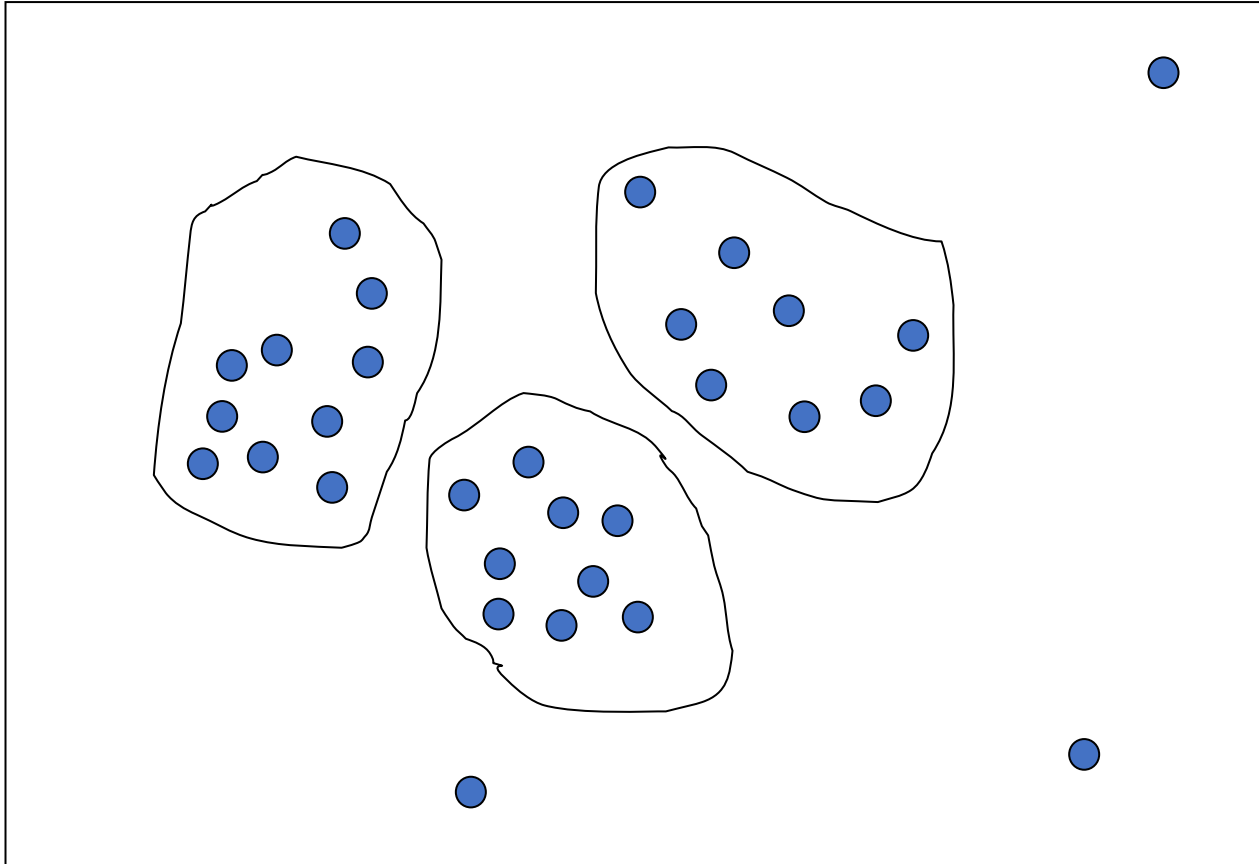
* Lấy theo means:

- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

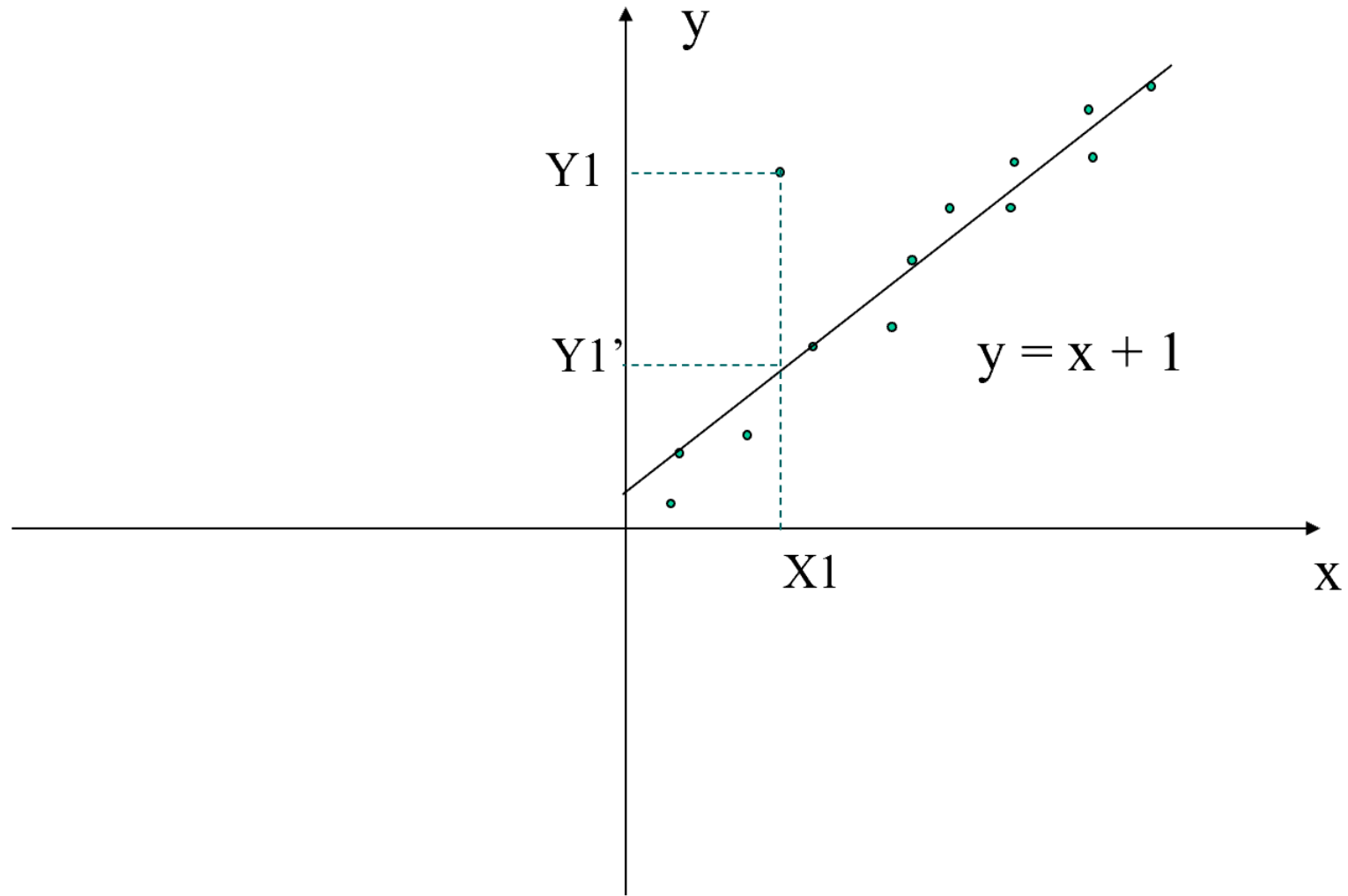
* Lấy theo biên:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Phân cụm



Hồi quy



LÀM SẠCH DỮ LIỆU

- **Bước 3: kiểm tra và sửa tính** không nhất quán trong dữ liệu.
 - Phát hiện kiểm tra thủ công sự bất thường trong giá trị dữ liệu
 - Dùng để sửa tính không nhất quán dữ liệu
 - Công cụ chà dữ liệu (Data scrubbing tools) : dùng cho một lĩnh vực cụ thể.
 - Công cụ kiểm toán dữ liệu (Data auditing tools) : dùng cho việc phân tích dữ liệu, xác định quan hệ, xác định các luật.

inconsistent: Chứa dữ liệu khác nhau của cùng codes hoặc names, e.g.,

- *Age="42", Birthday="03/07/2010"*
- Was rating "1, 2, 3", now rating "A, B, C"
- discrepancy between duplicate records

MỤC LỤC

- Đặt vấn đề
- Làm sạch dữ liệu
- Tích hợp
- Chuyển đổi
- Giảm chiều

2. Tích hợp dữ liệu (Data integration)

- Là bước kết hợp dữ liệu từ nhiều nguồn lại thành 1 nguồn duy nhất
- Tích hợp các bảng: e.g., $A.cust-id \equiv B.cust-\#$
 - Tích hợp metadata từ các nguồn khác nhau
- Vấn đề định danh entity:
 - Định danh entities từ nhiều nguồn khác nhau có thể khác nhau, e.g., Bill Clinton = William Clinton
- Xác định và giải quyết vấn đề xung đột giá trị
 - Cùng một entity, thuộc tính biểu diễn khác nhau trên các nguồn khác nhau
 - Lý do có thể: Khác biểu diễn, khác thang đo,...

2. Tích hợp dữ liệu (Data integration)

- Đương đầu với dư thừa dữ liệu khi tích hợp
 - Thuộc tính giống nhau có tên khác nhau từ các cơ sở dữ liệu khác nhau
 - Một thuộc tính có thể là kết quả sinh ra từ 1 thuộc tính khác trong 1 bảng
- Có thể phát hiện sử dụng phân tích tương quan (correlation analysis)
 - Correlation coefficient cho dữ liệu số liên tục
 - Chi-square test cho dữ liệu kiểu mục

Ví dụ: Correlation coefficient

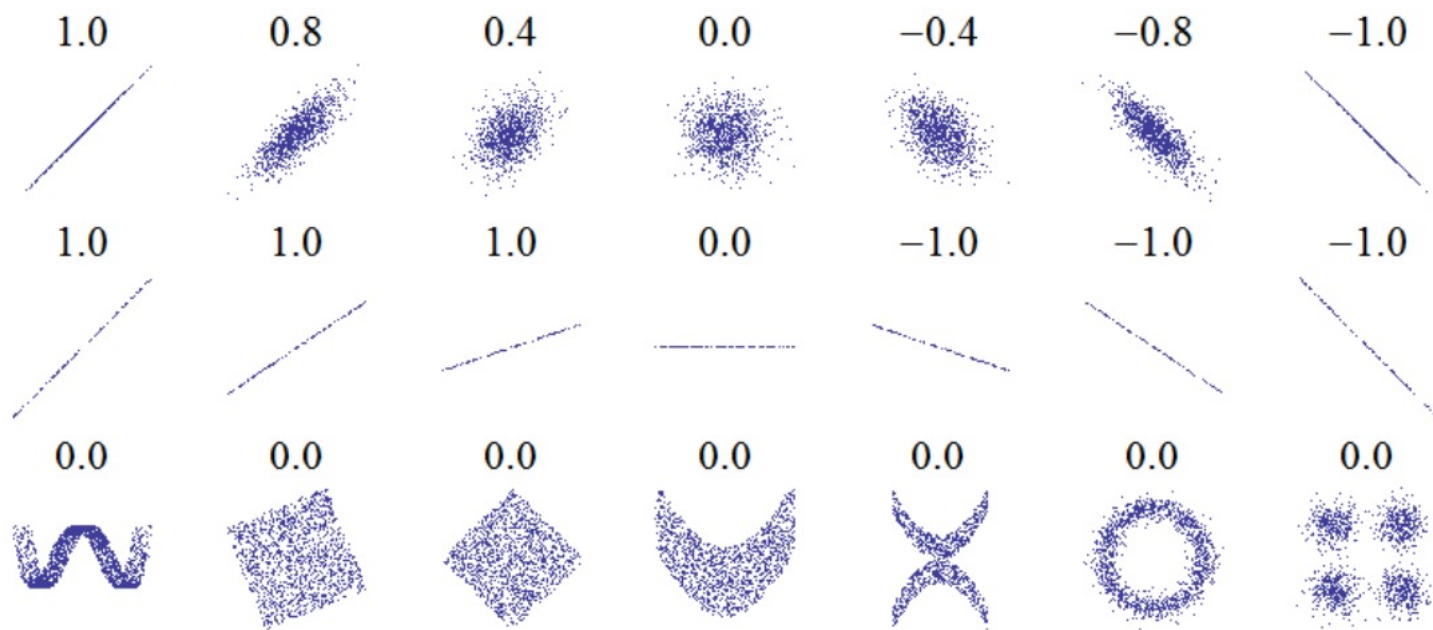
- Correlation coefficient (also called **Pearson's product moment coefficient**)

$$r_{A,B} = \frac{\sum_{i=1}^n (a_i - \bar{A})(b_i - \bar{B})}{(n-1)\sigma_A\sigma_B} = \frac{\sum_{i=1}^n (a_i b_i) - n\bar{A}\bar{B}}{(n-1)\sigma_A\sigma_B}$$

Với n là số bộ dữ liệu, \bar{A} và \bar{B} là kỳ vọng của các giá trị thuộc tính A và B , σ_A và σ_B là độ lệch chuẩn của A và B .

- Nếu $r_{A,B} > 0$, A và B có tương quan đồng biến.
- $r_{A,B} = 0$: độc lập; $r_{AB} < 0$: tương quan nghịch biến

Correlation coefficient



http://upload.wikimedia.org/wikipedia/commons/0/02/Correlation_examples.png

MỤC LỤC

- Đặt vấn đề
- Làm sạch dữ liệu
- Tích hợp
- Chuyển đổi
- Giảm chiều

3. Chuyển đổi (Data transformation and Data discretization):

- Là hàm chuyển đổi toàn bộ tập giá trị của thuộc tính cho trước tới một tập giá trị mới thay thế, trong đó mỗi giá trị cũ được thay thế bằng một giá trị mới.
- Phương pháp
 - Smoothing: Các phương pháp giảm nhiễu (binning, clustering, regression)
 - Xây dựng thuộc tính mới: Thuộc tính mới được xây dựng từ tập thuộc tính cũ
 - Tích hợp:
 - Chuẩn hóa:
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
 - Rời rạc hóa

Chuẩn hóa

- **Min-max normalization:** tới $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- Ví dụ: Khoảng thu nhập \$12,000 tới \$98,000 được chuẩn hóa thành $[0.0, 1.0]$. Khi đó, \$73,000 thành $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ : mean, σ : standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

- Ví dụ. $\mu = 54,000$, $\sigma = 16,000$. thì $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization thang 10**

$$v' = \frac{v}{10^j} \quad \text{Với } j \text{ là số nguyên bé nhất mà } \text{Max}(|v'|) < 1$$

MỤC LỤC

- Đặt vấn đề
- Làm sạch dữ liệu
- Tích hợp
- Chuyển đổi
- Giảm chiều

4. GIẢM CHIỀU

- **Ý nghĩa:** Việc giảm kích thước của dữ liệu cần đồng thời giữ được tính phân tích dữ liệu, tăng tốc quá trình khai phá/học máy

GIẢM CHIỀU

- Các chiến lược giảm kích thước dữ liệu
 - Lựa chọn tập con các thuộc tính : trong đó các thuộc tính không liên quan, dư thừa hoặc các chiều cũng có thể xóa hay loại bỏ
 - Giảm chiều : trong đó cơ chế mã hóa được sử dụng để giảm kích cỡ tập dữ liệu
 - Rời rạc hóa và trừu tượng khái niệm : trong đó các giá trị dữ liệu thô được thay thế bằng các khái niệm trừu tượng đã rời rạc hóa.

TỔNG KẾT

- Scikit-learn, pytorch, keras là những thư viện phổ biến sử dụng trong học máy
- Quy trình xây dựng hệ thống học máy: Thu thập dữ liệu, tiền xử lý, xây dựng mô hình, lựa chọn tham số và đánh giá mô hình, ứng dụng thực tế
- Vấn đề khi tiến hành thu thập dữ liệu dùng cho bài toán khai phá dữ liệu/học máy.
- Cần theo các bước của quy trình thu thập và tiền xử lý dữ liệu.
- Cần hiểu ý nghĩa trong từng bước của quy trình.



25 YEARS ANNIVERSARY
SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**



soict.hust.edu.vn/



fb.com/groups/soict

